A Literature Review on Supervised Machine Learning Algorithms and Boosting Process

M. Praveena
MCA, Mphil
Assistant Professor, Department of Computer Science, Dr. SNS Rajalakshmi College of Arts and Science, Coimbatore

V. Jaiganesh, PhD
Professor, Department of PG & Research
Dr. NGP Arts and Science College
Coimbatore

ABSTRACT
Data mining is one amid the core research areas in the field of computer science. Yet there is a knowledge data detection process helps the data mining to extract hidden information from the dataset there is a big scope of machine learning algorithms. Especially supervised machine learning algorithms gain extensive importance in data mining research. Boosting action is regularly helps the supervised machine learning algorithms for rising the predictive / classification veracity. This survey research article prefer two famous supervised machine learning algorithms that is decision trees and support vector machine and presented the recent research works carried out. Also recent improvement on Adaboost algorithms (boosting process) is also granted. From this survey research it is learnt that connecting supervised machine learning algorithm with boosting process increased prediction efficiency and there is a wide scope in this research element.

Keywords
Data mining, machine learning, research, adaboost, support vector machine, decision trees.

1. INTRODUCTION
Machine learning shortly describe as ML is a kind of artificial intelligence (AI) which compose available computers with the efficiency to be trained without being veraciously programmed. ML learning interest on the extensions of computer programs which is capable enough to modify when unprotected to new-fangled data. ML algorithms are broadly classified into three divisions namely supervised learning, unsupervised learning and reinforcement learning and is shown in Fig.1. The evolution of machine learning is comparable to that of data mining. Both data mining and machine learning consider or explore from end to end data to assume for patterns. On the other hand, in choice to extracting data for human knowledge as is the case in data mining applications; machine learning generate use of the data to identify patterns in data and fine-tune program actions therefore.

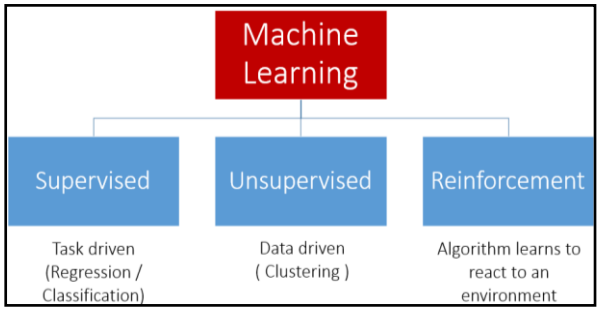

Fig.1. Machine Learning and its Types

Supervised machine learning is the mission of conceive a meaning from labelled training data which has a set of training examples. As far as supervised learning is concerned, every example is a mainstay containing an input object (which is usually a vector quantity) and a enforced output value (may also be referred as supervisory signal).

A supervised learning algorithm at first performs the analysis task from the practice data and constructs a contingent function, in order to map new examples. A maximum setting probably facilitates the algorithm to exactly courage the class labels for covered instances and the same needs the supervised learning algorithm to reduce from the training data to covered situations in a "rational" manner. The supervised methods are possibly used in various application areas that include marketing, finance, manufacturing, testing, stock market prediction, and so on.

1.1 Steps performed in the Supervised Machine Learning Algorithms
Step – 1: Establish the type of training examples. The user needs to courage the type(s) of data that will be used as a training set.

Step – 2: Converge a training set. The training set ambition to be delegate of the real-world use of the function. As a effort, a set of input objects is collected that remains and analogous outputs are also collected.

Step – 3: Resolve the input feature illustration of the learned function / learned attribute. The accurateness of the learned function is securely based on the input object is representation.

Step – 4: Resolve the formation of the learned function and comparable machine learning algorithm.

Step – 5: Assimilate the design and execute the learning algorithm on the collected training set.

Step – 6: Evaluate the accurateness / correctness of the learned function. Then, parameter adapt and learning may be performed on the resulting function and needs to be measured on a test data set that is break up from the training set.

1.2 Factors to be considered
1.2.1 Data Heterogeneity:
When the countenance vectors contains countenance of several kinds which includes discrete, discrete ordered, counts, continuous values, certain algorithms are simpler to implement than rest of the algorithms. Many such algorithms namely - Support Vector Machines, linear regression, logistic regression, neural networks, and nearest neighbour methods, desire that the input countenance be numerical and scaled to similar ranges.

1.2.2 Data Redundancy:

When the input features has unwanted information, a few learning algorithms probably may execute defectively due to numerical irresolution. Such researches issues may be solved by consolidate some pre-processing techniques.

1.2.3 Presence of interactions and non-linearity's:

When the countenance makes an autonomous role to the output, then algorithms that are based on linear functions and distance functions usually perform fit. On the other hand, when there are multifaceted interactions amongst countenance, then certain algorithms perform much better, as they are distinctively designed to determine these interactions.

2. RELATED WORKS

2.1 Recent Works on Decision Trees

Lertworaprachaya et al., 2014 [1] proposed a new model for compose decision trees using interval-valued fuzzy membership values. Most existing fuzzy decision trees do not consider the concerned associated with their membership values; however, precise values of fuzzy membership values are not always possible. Because of that, the authors represented fuzzy membership values as distance to model concerned and employ the look-ahead based fuzzy decision tree induction method to construct decision trees. The authors also measured the significance of different neighbourhood values and define a new parameter unkind to specific data sets using fuzzy sets. Some examples are provided to establish the effectiveness of their approach.

Bahnsen et al. 2015 [2] proposed an example-reliant cost-sensitive decision tree algorithm, by incorporating the different example-reliant costs into a new cost-based impurity measure and new cost-based pruning criteria. Subsequently, using three different databases, from three real-world applications namely credit card fraud detection, credit scoring and direct marketing, the authors evaluated their proposed method. Their results showed that their proposed algorithm is the best performing design for all databases. Additionally, when compared across a standard decision tree, their design builds significantly smaller trees in only a fifth of the time, while having a superior performance measured by cost savings, leading to a design that not only has more business-oriented results, but also a design that creates simpler models that are easier to analyze.

Online decision trees from data current are usually unable to handle concept drift. Blanco et al., 2016 [3] proposed the Incremental Algorithm Driven by Error Margins (IADEM-3) that mainly carry out two actions in response to a approach drift. At first, IADEM-3 resets the variables affected by the change and maintains unbroken the structure of the tree, which allows for changes in which ensuing target functions are very similar. After that, IADEM-3 creates alternative models that replace parts of the main tree when they significantly improve the accuracy of the model, thereby rebuilding the main tree if needed. An online change detector and a non-parametric statistical test based on Hoeffding's bounds are used to guarantee that significance. A new pruning method is also incorporated in IADEM-3, making sure that all split tests previously installed in decision nodes are useful. Their learning model is also viewed as an ensemble of classifiers, and predictions of the main and alternative models are joined to classify unlabeled examples. IADEM-3 is empirically related with various well-known decision tree induction algorithms for concept drift detection. The authors portrayed that their new algorithm generally reaches higher levels of accuracy with smaller decision tree models, maintaining the processing time bounded, irrespective of the number of instances processed.

Predicting learning styles in conversational intelligent tutoring systems using fuzzy decision trees has been proposed by Crokett et al., 2017 [4]. Prediction of learning style is carried out by imprison independent behaviour variables during the tutoring observation with the highest value variable. A weakness of their approach is that it does not take into consideration the interactions between behaviour variables and, due to the uncertainty inherently present in modelling learning styles, small differences in behaviour can lead to incorrect predictions. Subsequently, the learner is presented with guidance material not suited to their learning style. Because of the above mentioned challenges a new method that uses fuzzy decision trees to build a series of fuzzy predictive models connecting these variables for all dimensions of the Felder Silverman Learning Styles model. Results using live data by the authors showed that the fuzzy models have increased the anticipate accuracy across four learning style dimensions and facilitated the discovery of some interesting relationships amongst behaviour variables.

2.2 Recent Works on Support Vector Machine (SVM)

Motivated by the KNN trick conferred in the weighted twin support vector machines with local information (WLTSVM), Pan et al., 2015 [5] proposed a novel K-nearest neighbour establish structural twin support vector machine (KNN-STSVM). By applying the intra-class KNN method, different weights are given to the samples in one class to enhance the structural information. For the other class, the expendable constraints are deleted by the inter-class KNN method to speed up the coaching process. For large scale problems, a fast clip algorithm is further introduced for increase of rate. Comprehensive experimental results on twenty-two datasets demonstrate the efficiency of their proposed KNN-STSVM.

It is noteworthy that existing structural classifiers do not balance structural information's relationships both intra-class and inter-class. Connecting the structural information with nonparallel support vector machine (NPSVM), D. Chen et al. 2016 [6], designed a new structural nonparallel support vector machine (called SNPSVM). Each model of SNPSVM examine not only the concentration in both classes by the structural information but also the reparability between classes, thus it can fully adventure prior knowledge to directly recover the algorithms generalization capacity. Moreover, the authors applied the improved alternating direction designed of multipliers (ADMM) to SNPSVM. Both their model itself and the solving algorithm can guarantee that it possibly would deal with large-scale classification problems with a huge number of occurrence as well as features. Experimental results show that SNPSVM is superior to the other current algorithms based on structural information of data in both estimation time and classification accuracy.

Peng et al., 2016 [7] formulated a linear kernel support vector machine (SVM) as a consistent least-squares (RLS) problem. By defining a set of indicator variables of the errors, the solution to the RLS problem is represented as an equation that describe the error vector to the indicator variables. Through partitioning the training set, the SVM weights and tendency are expressed analytically using the support vectors. The authors also determine how their approach naturally extends to sums with nonlinear kernels whilst deflect the need to make use of Lagrange multipliers and duality theory. A fast constant solution algorithm based on Cholesky decomposition with

modification of the support vectors is recommended as a solution method. The properties of their SVM formulation were analyzed and correlated with standard SVMs using a simple example that can be decorated graphically. The correctness and behaviour of their proposed work has been demonstrated using a set of public benchmarking problems for both linear and nonlinear SVMs.

Utkin and Zhuk, 2017 [8] proposed a well-known one-class classification support vector machine (OCC SVM) dealing with interval-valued or set-valued training data. Their key idea is to represent every distance of training data by a finite set of explicit data with imprecise weights. Their representation is based on replacement of the interval-valued familiar risk produced by interval-valued data with the interval-valued expected risk produced by uncertain weights or sets of weights. It can also be mentioned that, the interval concern is replaced with the uncertain weight or probabilistic uncertainty. The authors showed how constraints for the imprecise weights are incorporated into dual quadratic programming problems which can be viewed as extensions of the well-known OCC SVM models. With the help of numerical examples with synthetic and real interval-valued training data the authors decorate their proposed approach and investigate its properties.

2.3 Recent Works on Adaboost

Universum data usually does not belong to any class of the training data, has been applied for training better classifiers. Xu et al., 2014 [9] addressed a novel boosting algorithm called UAdaBoost which possibly would better the classification performance of AdaBoost with Universum data. UAdaBoost determine a function by minimizing the loss for labelled data and Universum data. The cost function is discount by a greedy, stage wise, functional gradient procedure. Each training stage of UdaBoost is fast and efficient. The standard AdaBoost weights labelled samples over training iterations while UAdaBoost gives an explicit weighting program for Universum samples as well. Also the authors described the practical conditions for the effectiveness of Universum learning. These conditions are based on the analysis of the distribution of ensemble forecasting over training samples. By their experimental results the authors declare that their method can obtain superior performances over the standard AdaBoost by selecting proper Universum data.

Sun et al., 2016 [10] quoted a representative approach named noise-detection based AdaBoost (ND_AdaBoost) in order to improve the robustness of AdaBoost in the two-class classification scenario. In order to resolve the dilemma a robust multi-class AdaBoost algorithm (Rob_MulAda) is proposed by the authors whose key ingredients consist in a noise-detection based multi-class loss function and a new weight updating scheme. The authors claims that their experimental study indicates that their newly-proposed weight updating scheme is indeed more robust to mislabelled noises than that of ND_AdaBoost in both two-class and multi-class scenarios. As well, through the comparison experiments, the authors also verified the effectiveness of Rob_MulAda and provide a suggestion in choosing the most appropriate noise-alleviating approach according to the concrete noise level in practical applications.

Baig et al., 2017 [11] presented a boosting-based method of learning a feed-forward artificial neural network (ANN) with a single layer of hidden neurons and a single output neuron. At first, an algorithm called Boost on is depicted which learns a single-layer perception using AdaBoost and decision stumps. It is then extended to learn weights of a neural network with a single hidden layer of linear neurons. At last, a novel method is introduced by the authors to incorporate non-linear activation functions in artificial neural network learning. Their proposed method uses series representation to approximate non-linearity of activation functions, learns the coefficients of nonlinear terms by AdaBoost which adapts the network parameters by a layer-wise iterative traversal of neurons and an appropriate reduction of the problem. Comparison of various neural network models learned the proposed methods and those learned using the least mean squared learning (LMS) and the resilient back-propagation (RPROP) is provided by the authors.

Miller and Soh 2015 [12] proposed a novel cluster-based boosting (CBB) approach to address limitations in boosting on supervised learning (SL) algorithms. Their CBB approach partitions the training data into clusters containing highly similar member data and integrates these clusters directly into the boosting process. Their CBB approach attempts to address two specific limitations for current boosting both resulting from boosting focusing on incorrect training data. The first one is filtering for subsequent functions when the training data contains troublesome areas and/or label noise; and the second one is over fitting in subsequent functions that are forced to learn on all the incorrect instances. The authors demonstrated the effectiveness of CBB through extensive empirical results on 20 UCI benchmark datasets and proclaimed that CBB achieves superior predictive accuracy that use selective boosting without clusters.

3. FINDINGS AND CONCLUSIONS

Every learning algorithm will tend to suit some problem types better than others, and will typically have many different parameters and configurations to be adjusted before achieving optimal performance on a dataset, AdaBoost (with decision trees as the weak learners) is often referred to as the best out-of-the-box classifier. When used with decision tree learning, information gathered at each stage of the AdaBoost algorithm about the relative 'hardness' of each training sample is fed into the tree growing algorithm such that later trees tend to focus on harder-to-classify examples.

The supervised machine learning algorithms such as decision trees and support vector machine are capable enough to deal with big data mining tasks. Even though the algorithms efficiency considerably improving there is a need for adaptive boosting process required in order to increase the predictive accuracy much more. The following are the findings from this survey research manuscript.

(i) Fuzzy logic which is a soft computing technique is incorporated with the decision tree machine learning algorithm in order to rule out the ambiguity in the datasets.
(ii) Example – dependent along with cost sensitive factors helps the decision trees to proclaim more independency in machine learning process.
(iii) Error margins based methods reduce the false negative values while making use of decision trees.
(iv) Interactions between behaviour variables tend to improve the performance of the decision trees.
(v) Weight based structural information helps the support vector machine to quickly train the machine learning algorithm.

(vi) Relationships between inter-class and intra-class surely will increase the effectiveness of the support vector machine.
(vii) Decomposition of the attributes also significantly improves the effectiveness of the classifier.
(viii) Noise detection process will be helpful to increase the accuracy of the machine learning algorithm.
(ix) Cluster based boosting still has further scope of research by making use of optimization techniques.

From the above findings it is interesting to note that the clustering or classification accuracy directly depends on the employment of boosting process. Not only that the overall computational complexity would be reduced then. This survey research article chooses two machines learning algorithm and one boosting technique and portrayed on the recent research works carried out during 2014 to 2017.

4. FUTURE SCOPE OF RESEARCH

Dealing with several datasets and performing data mining is a tedious task. The following are the future directions for further research work.

- Optimization techniques like genetic algorithm, particle swarm optimization, ant colony optimization, artificial bee colony algorithms can be used for improving the performance of adaboost algorithm.
- Other machine learning algorithms such as relevance vector machine, extreme learning machine, neural networks can be used for classifying / clustering the data.

5. REFERENCES

[1] Y. Lertworaprachaya, Y. Yang, R. John, "Interval-valued fuzzy decision trees with optimal neighbourhood perimeter," Applied Soft Computing, vol. 24, pp. 851-866, 2014.

[2] A. C. Bahnsen, D. Aouada, B. Ottersten, "Example-dependent cost-sensitive decision trees," Expert Systems with Applications, vol. 42, pp. 6609-6619, 2015.

[3] F. Blanco, J. C. Ávila, G. R. Jiménez, A. Carvalho, A. O. Díaz, R. M. Bueno, "Online adaptive decision trees based on concentration inequalities," Knowledge-Based Systems, vol. 104, pp. 179-194, 2016.

[4] Crockett, A. Latham, N. Whitton, "On predicting learning styles in conversational intelligent tutoring systems using fuzzy decision trees," International Journal of Human-Computer Studies, vol. 97, pp. 98-115, 2017.

[5] X. Pan, Y. Luo, Y. Xu, "K-nearest neighbour based structural twin support vector machine," Knowledge-Based Systems, vol. 88, pp. 34-44, 2015.

[6] D. Chen, Y. Tian, X. Liu, "Structural nonparallel support vector machine for pattern recognition," Pattern Recognition, vol. 60, pp. 296-305, 2016.

[7] X. Peng, K. Rafferty, S. Ferguson, "Building support vector machines in the context of regularized least squares," Neurocomputing, volume. 211, pp. 129-142, 2016.

[8] V. Utkin, Y. A. Zhuk, "An one-class classification support vector machine model by interval-valued training data," Knowledge-Based Systems, vol. 120, pp. 43-56, 2017.

[9] J. Xu, Q. Wu, J. Zhang, Z. Tang, "Exploiting Universum data in AdaBoost using gradient descent," Image and Vision Computing, vol. 32, pp. 550-557, 2014.

[10] B. Sun, S. Chen, J. Wang, H. Chen, "A robust multi-class AdaBoost algorithm for mislabelled noisy data," Knowledge-Based Systems, vol. 102, pp. 87-102, 2016.

[11] M. Baig, M .M. Awais, E. M. El-Alfy, "AdaBoost-based artificial neural network learning," Neurocomputing, vol. 16, pp. 22 – 41, 2017.

[12] L. D. Miller and L. K. Soh, "Cluster-Based Boosting," IEEE Transactions on Knowledge and Data Engineering, vol. 27, pp. 1491-1504, 2015.